Extreme Learning Machine with Multi-Agent System for Regression

Chong Tak Yaw, Keem Siah Yap, Shen Yuong Wong, Chin Hooi Tan

Abstract: From the point of learning speed as well as generalization, Extreme Learning Machine (ELM) is widely known as an effective learning algorithm than the conventional learning methods. Basically, hidden neurons are not required in neuron alike, instead, weight is the parameter that would need to learn about the link in between output and hidden layers. The creation of an output is to integrate each independent of several ELMs. The precise approach is included in a Multi-Agent System. The novelty of ELM-MAS (Extreme learning machine based multi-agent system) is put forward in the paper for solving data regression problems. The ELMs consist of two layers which are the parent agent layer and individual agent layer. The effectiveness of the ELM-MAS model is proved with some activation functions employing benchmark datasets (abalone, strike and space-ga) and real world application (Nox, emission). The outcomes indicate that the proposed model is capable to attain improved results than other approaches.

Keywords: Extreme Learning Machine (ELM); Multi Agent System (MAS); Data Regression; NO, Emission of Power Plant

1. INTRODUCTION

Nowadays, ELM (known as Extreme Learning Machine) is well recognized as an effective learning algorithm from the point of generalization and learning speed as compared to the conventional learning approaches [1-7]. In addition, the universal approximation with input weights and haphazard biases is capable for ELM [8].

ELM (based Huang et al. [9, 10]) is exceptionally incline and effective to global optimum in the comparison to the conventional FNN (feedforward neural network). Furthermore, the greatest generalization bound of the conventional FNN is able to accomplish by ELM, in which every parameter in the state of normalization is learned with activation functions [11]. Based on the diverse types of problems [1-7], ELM performs much better than that of traditional FNN. There are a lot of fields that ELM have been applied in, for example biomedical analysis [12, 13], hyperspectral images [14], power systems [15, 16], action recognition [17], chemical process [18], system modelling [19, 20], and others.

There is some researches integration with every independent estimate of several ELMs to produce an output [21-25]. The kind of particular approach is the same as MAS (known as Multi Agent System) [26]. There are some researches that successfully applied for tackling problems in different domains, for example in health care [27-29], eCommerce [26, 30-32], military support [33, 34], knowledge management [35-38], decision support [39-42], as well as control systems [43-46]. The Fig. 1 is shown that the MAS’s general structure, in which the base platform is the collection of ELMs known as individual agents. Generally, an individual ELM (i.e. individual agent,)’s outcome is delivered to a place called parent agent, which is the concluding combined module.

In general, the mean output in the concluding combined module is derived from the approaches including weighted mean [7], voting [47], confusion matrix [48] and exact mean [22]. Regrettably, in the concluding combined module, these aforementioned approaches often need additional algorithms to generate outcome. In this paper, concluding combined module based on ELM is proposed.

A common method called meta-learning is used to collect and combine the results of several learners [49, 50]. This can roughly be defined as knowledge material by at least a learner [49, 50]. The model is achieved by multiple ELMs as hidden neurons to estimate outputs which are learned by the meta-learner. Theoretical investigation and experimental outcomes from several studies using the datasets of benchmark and artificial regression show that the Meta-ELM, which is trained by multiple ELMs, could give good performance at the expense of a lower computational cost [51]. Nonetheless, in this paper, an extreme learning machine-based multi agent system (ELM-MAS) is designed from alternative perception. For this proposed model, it has two layers of full ELMs: the first layer consists of a single ELM and it is the parent agent; the second layer consisted of at least an ELMs where each ELM is considered as an individual agent. This proposed ELM-MAS model with two layers’ structure resembles a typical MAS (Figure 1).
Extreme Learning Machine with Multi-Agent System for Regression

II. THE ALGORITHMS OF ELM AND ELM-MAS

An ELM can either be a RBF or a feedforward network with a sophisticated learning algorithm because it’s depending on the type of activation function it utilized as shown in Fig. 2. For example, \((x_j, t_j)\), i.e. \(x_j \in \mathbb{R}^M\) (\(M\) is quantity of input attributes), \(t_j \in \mathbb{R}^C\) (\(C\) is quantity of classes) and a series of \(N\) training samples (with a input vector as well as individual target output vector), is utilized to train an ELM consisting of \(L\) number of hidden neurons. As for the model in this paper, five ELMs are generated as individual agents, each had different random input weights and the class is set to 1. As shown in Figure3, the output of each ELM\(^k\) (for \(k = 1, 2, \ldots, 5\)), in response to \(x_j\) is

\[
\text{ELM}^k(x_j) = \sum_{i=1}^{L} \beta^k_i G(a^k_i, b^k_i, x_j) = t_j
\]

(1)

for \(j = 1, \ldots, N\) and for \(c = 1, \ldots, C\)

Where \(\beta^k_i\) is the output weights, \(a^k_i\) and \(b^k_i\) are the bias and input weights (learning constraints) of the hidden neurons, whereas \(G(a^k_i, b^k_i, x_j)\) is the output of the \(i^{th}\) hidden neuron given the input vector \(x_j\).

The following equations respectively reveal the description of the \(G(a^k_i, b^k_i, x_j)\) for additive sigmoid hidden neuron as well as RBF hidden neuron.

\[
G(a^k_i, b^k_i, x_j) = \frac{1}{1 + \exp\{- (a^k_i \cdot x_j + b^k_i)\}}.
\]

(2)

\[
b^k_i \in R
\]

\[
G(a^k_i, b^k_i, x_j) = \exp\{- b^k_i \|x_j - a^k_i\|^2\}
\]

(3)

\[
b^k_i \in R^C
\]

![Fig. 1 A General Structure of Multi Agent System (MAS)](image1)

Fig. 1 A General Structure of Multi Agent System (MAS)

![Fig. 2 The Construct of ELM](image2)

Fig. 2 The Construct of ELM

III. THE METHODOLOGY OF ELM-MAS

The procedure of the training phase is detailed below.

Stage 1: Randomly allocate the input weights \(a^k_i\) and \(b^k_i\) for \(k = 1, 2, \ldots, 5\) and \(i = 1, \ldots, L\).

Stage 2: The hidden layer output matrix for ELM\(^k\) (for \(k = 1, 2, \ldots, 5\)), \(H^k\), is calculated as follows.

\[
H^k = \begin{bmatrix} G(a^k_1, b^k_1, x_1) & \cdots & G(a^k_L, b^k_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a^k_1, b^k_L, x_N) & \cdots & G(a^k_L, b^k_L, x_N) \end{bmatrix}_{N \times L}
\]

(4)

Stage 3: Calculation of the \(\beta^k\) (output weights of ELM\(^k\)). Because of that \(H\) is possibly a non-symmetrical matrix, the inverse matrix cannot be solved. So to tackle this problem, the Moore-Penrose pseudo inverse matrix method is adopted as following equation.

\[
\beta^k = \left( (H^k)^T (H^k) \right)^{-1} (H^k)^T T,
\]

(5)

where \(T = [t_1, \ldots, t_N]^T\) is corresponding targeted output vectors.

Stage 4: After output weights of ELM\(^k\) are calculated, the outputs of ELM\(^k\) are computed using the training samples.

\[
y^k = \text{ELM}^k(x_j) = \sum_{i=1}^{L} \beta^k_i G(a^k_i, b^k_i, x_j)
\]

(6)

for \(j = 1, \ldots, N\) and for \(c = 1, \ldots, C\).
Stage 5: Randomly allocate the \( p_i \) and \( q_i \) (input weights for parent ELM), for \( i = 1, \ldots, L_i \), where \( L_i \) is referring to the quantity of hidden neurons of parent ELM.

Stage 6: Computation of \( S \) (hidden layer output matrix for parent ELM), as follow

\[
S = \begin{bmatrix}
G(p_1, q_1, w_1) & \ldots & G(p_{L_1}, q_{L_1}, w_1) \\
\vdots & \ddots & \vdots \\
G(p_1, q_1, w_N) & \ldots & G(p_{L_1}, q_{L_1}, w_N)
\end{bmatrix}_{N \times L_1}
\]  

(7)

Where \( w_j \) is the summation outputs of ELM\(^k\) (for \( k = 1, 2, \ldots, 5 \)) in response to \( x_j \), i.e., \( w_j = [y_1^j, y_2^j, y_3^j, y_4^j, y_5^j] \), \( y_j^k \in \mathbb{R}^C \), and \( w_j \in \mathbb{R}^{5C} \).

Stage 7: Determine the output weights of parent ELM, \( \alpha \) by the equation below,

\[
\alpha = \left( S^T S \right)^{-1} S^T T
\]

(8)

where \( T = [t_1, \ldots, t_N]^T \) is the corresponding targeted output vectors.

As soon as every sample data is trained with Stage 1-7, the ELM-MAS is utilized for validation of an new input vector \( z \) established on the \( a_k^L, b_k^L, p, q, p_k^L \) and \( \alpha \), i.e.,

\[
h_k^L = [G(a_k^L, b_k^L, z) \ldots G(a_k^L, b_k^L, z)]_{1 \times L_1}
\]

(9)

\[y_k^L = h_k^L p_k^L\]

(10)

\[s = [G(p_1, q_1, v) \ldots G(p_{L_1}, q_{L_1}, v)]_{1 \times L_1}
\]

(11)

\[\bar{y} = s \alpha
\]

(12)

Where \( h^k \) and \( y^k \) are hidden layer and output of ELM\(^k\) respectively, \( v = [v_1, v_2, v_3, v_4, v_5] \) is combination outputs of the ELM\(^k\) in response to \( z \), whereas \( s \) and \( \bar{y} \) are hidden layers of final output of the validation respectively.

After computing the output of ELM for testing samples, the RMSE (root mean squared error) is computed.

### IV. EXPERIMENTAL AND RESULTS USING BENCHMARK DATA

There are three benchmark datasets (namely Abalone, Strike and Space-ga) from the UCI machine source which are used to test the capability of ELM-MAS. The specifications of datasets are detailed in Table 1 as shown below. All experimentations are run in MATLAB (version 2010) on a private laptop armed with 8G RAM and Intel(R) Core(TM) i7 2.9 GHz CPU.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>No. of Attributes</th>
<th>No. of Training Samples</th>
<th>No. of Testing Samples</th>
<th>No. of Total Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abalone</td>
<td>8</td>
<td>3000</td>
<td>1177</td>
<td>4177</td>
</tr>
<tr>
<td>Strike</td>
<td>6</td>
<td>416</td>
<td>209</td>
<td>625</td>
</tr>
<tr>
<td>Space-ga</td>
<td>6</td>
<td>2071</td>
<td>1036</td>
<td>3107</td>
</tr>
</tbody>
</table>

For the setting of the experiment, the quantity of hidden neurons of each ELM\(^k\) is fixed (i.e., \( L \)) to 180 for the three benchmark datasets. On the other hand, two-thirds of the training samples are used for training while the remaining one-third are utilized to work out the most suitable number of neurons of the parent ELM (i.e., \( L_i \)) via a validation process. The validation and training processes are started by setting \( L_i = 50 \) units and then increased by an increment of \( 50 \) units for each type of the activation function of ELM-MAS.

Table 2 outlines the summary of outcomes by means of ELM-MAS for all kinds of activation functions in the context of the RMSE and the quantity of hidden neurons. The best results (i.e., Laplace Basis for both abalone and strike, and Sigmoid for Space-ga) are presented in Table 2.

### Table 1 Prescription of Benchmark Datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>No. of Attributes</th>
<th>No. of Training Samples</th>
<th>No. of Testing Samples</th>
<th>No. of Total Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abalone</td>
<td>8</td>
<td>3000</td>
<td>1177</td>
<td>4177</td>
</tr>
<tr>
<td>Strike</td>
<td>6</td>
<td>416</td>
<td>209</td>
<td>625</td>
</tr>
<tr>
<td>Space-ga</td>
<td>6</td>
<td>2071</td>
<td>1036</td>
<td>3107</td>
</tr>
</tbody>
</table>

### Table 2 Sumarry of RMSE of the ELM-MAS with Different Activation Functions

<table>
<thead>
<tr>
<th>Activation Functions</th>
<th>Abalone</th>
<th>Strike</th>
<th>Space-ga</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>Hidden Neurons</td>
<td>RMSE</td>
</tr>
<tr>
<td>RBF</td>
<td>0.016859</td>
<td>20</td>
<td>0.037227</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>0.015363</td>
<td>4</td>
<td>0.124423</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.013751</td>
<td>4</td>
<td>0.048052</td>
</tr>
<tr>
<td>Laplace Basis</td>
<td>0.013973</td>
<td>9</td>
<td>0.041143</td>
</tr>
<tr>
<td></td>
<td>0.014845</td>
<td>11</td>
<td>0.018766</td>
</tr>
</tbody>
</table>

The outcomes of the proposed ELM-MAS are also compared to the results of other ELM-based methods. As observed from Table 3, smaller value of RMSE represents better results. Therefore, the RMSE of ELM-MAS are better as compared to SVM[51] and ELM[3].

### Table 3 RMSE of ELM-MAS, ELM [3] and SVM [51]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Abalone</th>
<th>Strike</th>
<th>Space-ga</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>RMSE</td>
<td>RMSE</td>
</tr>
<tr>
<td>ELM-MAS</td>
<td>0.0174</td>
<td>0.1637</td>
<td>0.0068</td>
</tr>
<tr>
<td>SVM [51]</td>
<td>0.0764</td>
<td>0.2282</td>
<td>0.0648</td>
</tr>
<tr>
<td>ELM [3]</td>
<td>0.0761</td>
<td>0.2985</td>
<td>0.0624</td>
</tr>
</tbody>
</table>
V. REAL WORLD APPLICATION: NOₓ EMISSION OF POWER GENERATION PLANT

In the context of real world application in this study, the NOₓ emission of an open cycle gas turbine in a power generation plant (located at Port Dickson, Malaysia) had been investigated [52]. The objective is to establish a neural network model for the prediction of NOₓ emission. There are 150 input attributes taken from the parameters of the power generation plant such as the loading of the gas turbine, temperature, pressure and etc. The targeted output is the quantity of NOₓ (in ppm) emitted from the gas turbine.

A total of 3,405 data samples had been collected for training and testing of ELM-MAS. Out of 3,405 data samples, 2,270 are used for training while left behind 1135 are used for testing as shown in Table 4. An experiment is conducted on the testing datasets for fifty runs and the average results are recorded. A summary result for NOₓ emission with different activation functions is shown in Table 5. The best RMSE result in Table 5 is 0.0049 using Sigmoid activation function.

Table. 4 Specification of NOₓ Emission Datasets [52]

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Attributes</th>
<th># Training</th>
<th># Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOₓ emission</td>
<td>150</td>
<td>2,270</td>
<td>1,135</td>
</tr>
</tbody>
</table>

Table. 5 Summary of ELM-MAS for NOₓ Emission Datasets with Different Activation Functions

<table>
<thead>
<tr>
<th>Activation Functions</th>
<th>NOₓ Emission</th>
<th>RMSE</th>
<th># Hidden Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>0.0061</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>Sigmoid</td>
<td>0.0049</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.0073</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>LaplaceAct</td>
<td>0.0064</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Laplace Basis</td>
<td>0.0058</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

In the experiment of using ELM, two-thirds of the data samples are used for training while the remaining one-third are used for the validation process to decide the most appropriate number of neurons of the parent ELM (i.e. L). The validation and training processes for sigmoid activation function (SigAct) of ELM are started by setting \( L = 50 \) units and then increased by an increment of 50 units. An experiment is conducted on the testing datasets for fifty runs and the average results are recorded. As an example, Table 6 showed the testing processes based on SigAct. Based on the results of RMSE (Table 6), the best RMSE obtained is 0.027086. The max and min of RMSE for \( L = 200 \) are 0.028416 and 0.026952 respectively. Using the result in Table 6 to compare with Table 5, the RMSE results of ELM is higher than RMSE of ELM-MAS due to the complexity of hidden neurons in ELM.

VI. CONCLUSIONS

In essence, a new two layers of ELMs model called ELM-MAS is established for regression. The ELM-MAS model is certified by consuming benchmark datasets for instance abalone, strike and space-ga. The experimental outcomes demonstrates that the proposed model is better preferable than SVM [51] and ELM [3], as shown in Table 3. Moreover, the ELM-MAS model is assessed by applying it on the real world application which is NOₓ emission of power generation plant. Thus far, our results demonstrated that the RMSE of ELM-MAS for NOₓ emission is better that ELM.

Although results obtained from the abalone, strike and space-ga as well as NOₓ emission are reassuring, advance study with datasets from diverse application fields is essential for validating the capability of ELM-MAS application in real world.

ACKNOWLEDGMENTS

This work was supported by Universiti Tenaga National under grant J510050684.

REFERENCES


44. Trappey, A.J., et al., Intelligent engineering asset management system for power transformer maintenance decision supports under various operating conditions. Computers & Industrial Engineering, 2015. 84: p. 3-11.


47. Jenkins, D.P., M. BOYD, and C. LANGLEY. Using the decision ladder to reach a better design. in The Ergonomics Society Annual Conference. 19e21 April. 2016.

48. Marom, N.D., L. Rokach, and A. Shmilovich, Using the confusion matrix for improving ensemble classifiers. in 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel. 2010. IEEE.


AUTHORS PROFILE

Chong Tak Yaw, Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Malaysia.

Keem Siah Yap, Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Malaysia.

ShenYuong Wong, Department of Electrical and Electronics Engineering, Xiamen University Malaysia, Malaysia.

Chin Hooi Tan, Business Innovation Incubator, Tenaga Nasional Berhad, Malaysia.